Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

ROLLING HORIZON SCHEDULING IN LARGE JOB SHOPS

Kristin A. Thoney
Jeffrey A. Joines
Padmanabhan Manninagarajan

College of Textiles
North Carolina State University
Box 8301
Raleigh, NC 27695, U.S.A.

Thom J. Hodgson

College of Engineering
North Carolina State University
Box 7906
Raleigh, NC 27695, U.S.A.

ABSTRACT

The Virtual Factory is a job shop scheduling tool that was developed at NC State. It has been shown to provide near-optimal solutions to industrial-sized problems in seconds through comparison to a computed lower bound. It is an iterative simulation-based procedure, whose objective is minimizing maximum lateness. Like many other job shop scheduling tools, the Virtual Factory has been evaluated primarily in a transient setting, even though a rolling horizon setting is more indicative of the situation in which scheduling algorithms are used in industry. Consequently, a rolling horizon procedure has been developed with which the Virtual Factory was tested. Experimental results indicate that the Virtual Factory also performs well under these circumstances.

1 INTRODUCTION

There are many tools available for scheduling job shop problems. The Virtual Factory, developed at NC State, is one such tool that has been found to provide near-optimal solutions to industrial-sized problems in seconds. The Virtual Factory is an iterative simulation-based procedure, that solves deterministic problems. Its objective is minimizing maximum lateness, L_{max}.

The Virtual Factory, as many of the job shop scheduling algorithms found in the literature, has been tested primarily under transient circumstances (similar to simulating a terminating system). In industry, though, running a plant until it is empty is rare. Instead, plants usually contain many different orders, with new orders arriving as older ones are completed. Scheduling is often performed on some regular basis, i.e. everyday. The best schedule is implemented until the plant is rescheduled. Thus, scheduling occurs on a rolling horizon basis. Rolling horizon scheduling has been discussed in the literature, but experimentation has concentrated primarily on lot sizing problems. To evaluate how well the Virtual Factory might perform in industry, it therefore will be tested in a rolling horizon setting.

In Section 2, an overview of the Virtual Factory is presented. Section 3 explains the rolling horizon version of the Virtual Factory that was implemented, as well as how rolling horizon problems were generated. Experimental results are found in Section 4. Section 5 provides the conclusions.

2 VIRTUAL FACTORY

The idea for this simulation-based job shop scheduling algorithm was first proposed by Lawrence and Morton (1986) and Vepsalainen and Morton (1988). Hodgson et al. (1998, 2000) further developed it and named it the Virtual Factory. The Virtual Factory consists both of a scheduling algorithm and a lower bound.

2.1 Scheduling Procedure

Let d_i be the due date of job i and p_{ij} be the processing time of job i on machine j. Then the slack of job i on machine m is calculated as

$$Slack_{i,m} = d_i - \sum_{j \in m+} p_{ij} \tag{1}$$

where $m+$ is the set of all operations subsequent to machine m on job i's routing. Slack represents the latest possible time that a job can finish on a machine and still satisfy its final due date. As this does not include queuing time, slack did not perform well as a dispatching rule in early experiments found in the scheduling literature.

To remedy this situation, a revised slack value that incorporates queuing times is used as the sequencing rule in the Virtual Factory. Queuing times are recorded for each job at each machine it visits in one iteration of the simulation and used in the next iteration. The revised slack for job i on machine m is computed as

$$Slack'_{i,m} = d_i - \sum_{j \in m+} p_{ij} - \sum_{j \in m++} q_{ij} \tag{2}$$

where m++ is the set of all subsequent operations to machine m on the routing sheet for job i, except the immediate subsequent operation. The simulation is run until the lower bound is achieved or a specified number of iterations is reached, and the best solution is saved.

2.2 Lower Bound

Hodgson et al. chose to evaluate the quality of the schedules produced by the Virtual Factory through comparison to a lower bound. The lower bound is calculated by decomposing the job shop problem into individual one machine problems. To do this, an earliest start time and a latest finish time are calculated for each machine on each job's route. Let r_i be the release time of job i. Then the earliest possible start time for a job i on machine m is,

$$ES_{i,m} = r_i + \sum_{j \in m-} p_{ij} \tag{3}$$

where m- is the set of all operations preceding machine m on job i's routing sheet. The latest finish time for each job i on machine m is

$$LF_{i,m} = d_i - \sum_{j \in m+} p_{ij} \tag{4}$$

where $m+$ is the set of all operations following machine m on the routing sheet of job i.

The lower bound for the job shop problem ($N/M/L_{max}$) is obtained by solving the $N/1/\ L_{max} \mid r_i$ problem on each machine m by considering $LF_{i,m}$ as the effective due date for job i on machine m and $ES_{i,m}$ as the release time (r_i) for job i on machine m. Since $N/1/\ L_{max} \mid r_i$ is NP-hard, a relaxation suggested by Baker and Su (1974) is used. The relaxation is to allow preemption of a job in process whenever one with a more imminent due date becomes available.

The overall lower bound, $LB\ (L_{max})$, is computed as

$$LB(L_{\max}) = \max_{m=1,M}\{LB_m(L_{\max})\} \tag{5}$$

where $LB_m\ (L_{max})$ is the lower bound for machine m. The power of this lower bound is that there are M chances to get a tight bound.

3 ROLLING HORIZON SCHEDULING

The following definitions are required for this section:

- t - Current time in days
- c_j - Completion time of job j
- N - Total number of jobs
- N_s - Total number of jobs starting in factory on first day
- M - Total number of machines
- UL - Upper limit of uniform distribution for number of operations
- JR - Number of jobs released each day
- RO - Number of operations for jobs released
- DL - Length of a day
- T - Total horizon length in days
- w - Number of days in warm-up period
- WIP - Work in process (number of days)
- i - Number of iterations

3.1 Scheduling Procedure

The algorithm for the rolling horizon scheduling procedure is given as follows:

```
1. Initialize t = 0
    1.1 If t = w + 1, compute LB
    1.2 Release jobs whose rj = t
    1.3 Run the Virtual Factory i iterations
    1.4 Implement the first day of the best
    schedule
    1.5 t = t + 1
    1.6 Continue from 1.1 until t = T
2. Run the remainder of the best schedule until all jobs are finished
3. Initialize j =1
    3.1 If cj > w, determine if job j is the
    Lmax job
    3.2 j = j + 1
    3.3 Continue from 3.1 until j = N
```

Step 1 initializes the beginning of the first day as time 0. If in step 1.1 the time is one day past the warm-up period, the lower bound is computed. In step 1.2, the jobs with release time equal to the current time enter the factory. No jobs are released on the beginning of the first day since these jobs are assumed to be already in the factory. Step 1.3 runs the original VF procedure for a fixed number of iterations. In step 1.4, the first day of the best schedule is implemented. The rest of the schedule is discarded, except on the last day. At the end of the day, there may be jobs that are still in process. Each of these jobs is put back in the machine's queue, and the job's processing time is set equal to the remaining processing time. Steps 1.5 and 1.6 ensure that steps 1.1 through 1.4 are run for each day until the total number of days is reached. In step 2, the best schedule is run until all jobs are finished. This ensures that the scheduling procedure does not sacrifice the remaining jobs in the factory to yield a good schedule for the jobs that complete processing since all jobs finish. Step 3 initializes the counter, *j*, equal to the first job. In Step 3.1, only jobs that are completed after the warm-up period are included in the L_{max} calculation to eliminate transient effects dependent on initial factory conditions. Steps 3.2 and 3.3 ensure that the lateness for each job completed after the warm-up period is compared to the current maximum lateness.

3.2 Lower Bound

The lower bound for the rolling horizon schedule is computed after the warm-up period. The *LB* calculation includes both jobs that are currently in the factory after the warm-up period, with their remaining operations and processing times, and also those jobs that will be released later, during the complete horizon of the simulation. The *LB* is computed in the same manner as for the original VF. Therefore, even though there are multiple runs of the VF engine for the rolling horizon scheduling procedure, there is only one *LB* calculation.

3.3 Problem Generation

For testing the Virtual Factory on a rolling horizon basis, several problems were generated. In each problem, *DL*=1600, *T*=100, and *i*=100. Initially, a *M* and *UL* value were specified. Then, *RO* and *WIP* were set equal to *UL*. For the 5 operation problem depicted in this paper, *M*=50 and *UL*=*RO*=*WIP*=5. Finally, the values of *JR*, N_s, and *w* were determined based on the other parameters of the problem.

To compute *JR*, the number of jobs that balances the input into the factory with the output from the factory needed to be found. This was approximated by dividing the average number of operations that can be processed daily by the number of machines on which the jobs released after the first day are processed. To find the average number of operations that can be processed each day in the factory, the number of machines, *M*, was multiplied by the average number of operations that a single machine can process in a day, $\overline{Mops}$. $\overline{Mops}$ can be computed by dividing the day length, *DL*, by the average processing time, $\overline{P}$. Consequently,

$$\overline{Mops} = \frac{DL}{\overline{P}} \tag{6}$$

and

$$JR \approx \frac{(M)(\overline{Mops})}{RO}. \tag{7}$$

Since the processing times for the problems are uniformly distributed between 1 and 200, $\overline{P} \approx 100$ and thus $\overline{Mops} \approx 1600/100=16$. (Assuming an 8 hour work day, the average processing time is 0.5 hours.) Therefore, for the 5 operation problem, $JR \approx (50)(16)/5=160$. This value tends to overestimate *JR* since it assumes that there is never any idle time on the machines. Therefore, experimentation was performed to determine the actual value of *JR*, starting with the computed value. *JR* was found to be 151 for the 5 operation problem.

N_s was computed to achieve the desired amount of *WIP*. Since the problems have been designed so the factory input is approximately equal to the factory output, the number of operations that will be completed each day is approximately (*JR*)(*RO*). If *WIP* days of work in process is desired, then the total number of operations that should start in the factory is (*JR*)(*RO*)(*WIP*). Each job that starts in the factory has an average of $\overline{Ops}$ operations, where $\overline{Ops} = (UL+1)/2$. Consequently,

$$N_s \approx \frac{(JR)(RO)(WIP)}{\overline{Ops}} \tag{8}$$

jobs should start in the factory. For the 5 operation problem, $N_s \approx [(151)(5)(5)]/3 \approx 1258$.

The length of the warm-up period was chosen to eliminate potential transient effects caused by the initial jobs in the factory. The warm-up period in days, *w*, was set equal to 10 since this is significantly larger than the *WIP* in the 5 operation problem.

4 EXPERIMENTATION

Due date range has been shown by Demirkol et al. (1998) to be a factor influencing solution performance in evaluating algorithms scheduling transient job shops. Due date range may also be a factor in scheduling jobs shops in rolling horizon scenarios. Therefore, experiments were run for due date ranges between 0 and 25 days. A due date range, *DDR*, is defined so that each job, *j*, is randomly generated a discrete uniform due date between r_j and $r_j + DDR$, where $r_j=0$ for jobs initially in the factory. Note that all jobs are released at the beginning of a day, whereas the due date for a job could occur at any time during the day. For each due date range, 20 replications were run and the average difference between L_{max} and *LB* was calculated. This difference is the maximum by which the simulation solution could exceed the optimal solution. A positive difference between L_{max} and *LB* could be the result of a non-optimal schedule, a weak *LB*, or a combination of the both.

4.1 Base Case

Results of the 5 operation problem can be seen in Figure 1. The average *LB*-L_{max} is approximately in the range of 0.9 to 0.18 days. For the first 14 due date ranges, the average difference does not exceed 0.15 days. There is a slight increase in the differences for due date ranges beyond 13 days. These differences are quite small considering that 90 days of factory performance was included in these statistics, with the latenesses of over (90)(151)=13,590 jobs taken into account. This indicates that the scheduling procedure is performing well.

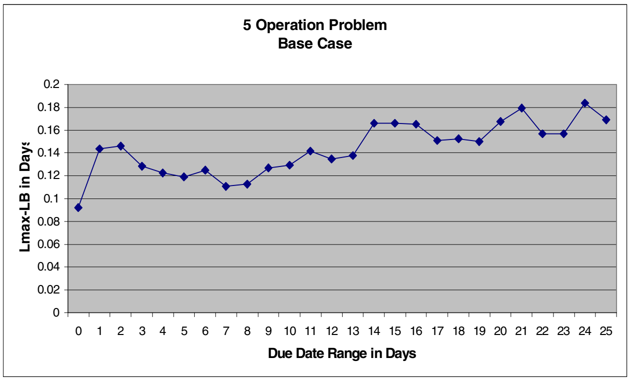

Figure 1: Base Case

4.2 Varying the Total Horizon Length

To determine the effect of the total number of days that are scheduled on the quality of the scheduling solutions, each problem was run for 55 days and 190 days with the same 10 day warm-up. This allows the scheduling solutions to be observed when the total number of days after the warm-up period are half as many and twice as many as in the base case.

The results of the 5 operation problem with 55 days is shown in Figure 2. The average L_{max}- *LB* value is low for due date ranges up to 13 days. Then it jumps up and the results are similar to that of the base case. This indicates that for large due date ranges, the differences between L_{max} and *LB* do not change much, on average, between 55 and 100 days, but they do increase significantly for small due date ranges.

The results for running the 5 operation problem for 190 days can be seen in Figure 3. The performance has gotten somewhat worse as the total horizon length has been increased. The maximum average difference between L_{max} and *LB*, though, is still only about 0.26 days.

4.3 Increasing the Warm-Up Period

The problems were run with an increased warm-up period to test if the transient effects were, indeed, eliminated. Figure 4 shows the comparison of increasing the warm-up period with the base case. Increasing the warm-up period yielded slightly better results for due date ranges up to 17 days. It is difficult to determine if this difference indicates that there are some transient effects remaining when a warm-up period of 10 days is used or if this is a result of the slight decrease in performance as the horizon length is increased, evidenced in Section 4.2.

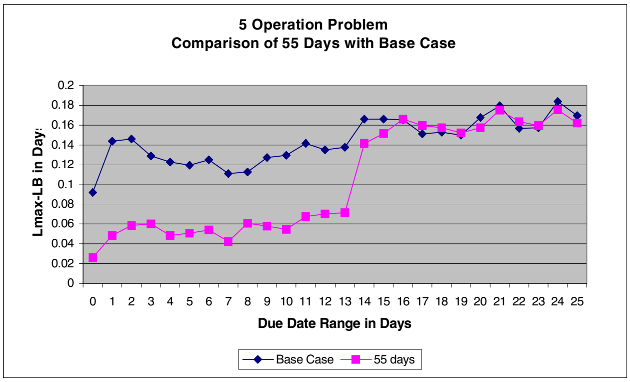

Figure 2: Decreasing the Total Horizon Length

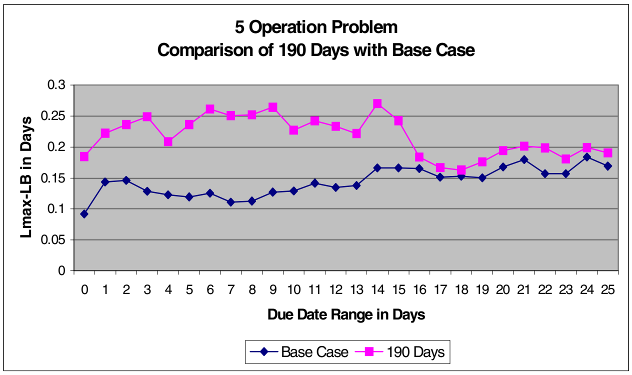

Figure 3: Increasing the Total Horizon Length

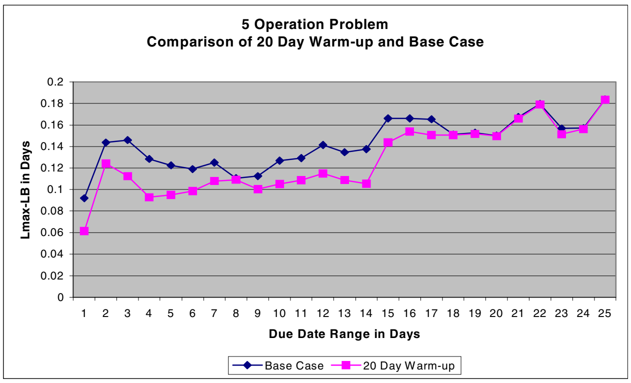

Figure 4: Increasing the Warm-up Period

4.4 Varying the Number of Jobs Released

In industry, it would be uncommon for a factory to release exactly the same amount of jobs each day. Thus, to see the impact that varying the number of jobs released each day has on the ability of the VF to provide good schedules, experiments were carried out in which the average number of jobs released was approximately equal to the number released in the base cases. For the 5 operation problem, the number of jobs released each day was uniformly distributed between 145 and 155. Figure 5 shows that there is little difference between the base case and the corresponding case where the operations were varied.

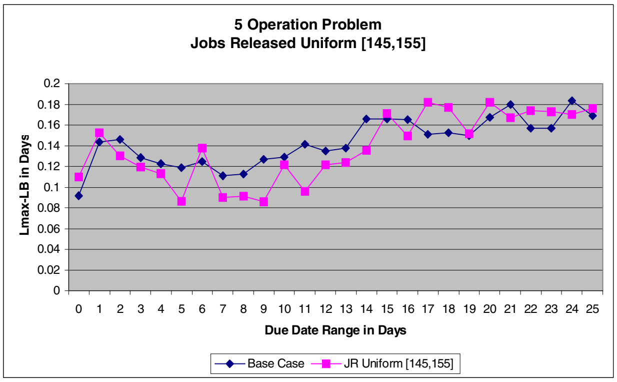

Figure 5: Varying Number of Jobs Released Each Day

4.5 Varying the Number of Operations

Releasing jobs with varying number of operations is also a typical occurrence in industry that has the potential to effect the performance of a scheduling algorithm. Therefore, this parameter has been varied, setting the average number of operations equal to the number of operations used in the base cases. For the 5 operation problem, the number of operations remaining for the jobs released each day was varied uniformly between 3 and 7. Figure 6 shows the comparison of changing the number of operations with the base case. When compared with the base case, varying the number of operations yields slightly worse results for high due date ranges.

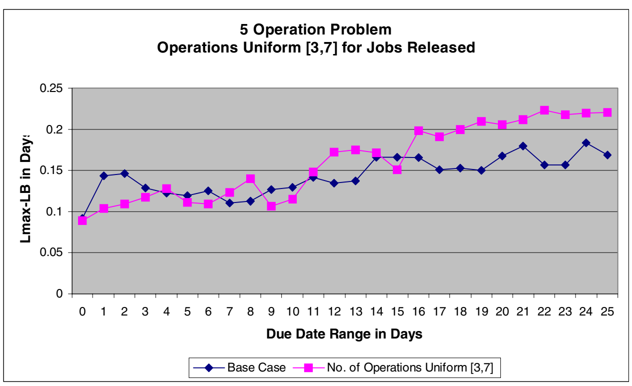

Figure 6: Varying Number of Operations for Jobs Released

4.6 Varying the Number of Jobs Released and the Number of Operations

Since both releasing different numbers of jobs per period and releasing jobs with varying numbers of operations is common in industry, these variations should be also tested simultaneously.

Figure 7 shows the comparison of changing both the number of jobs released and the number of operations remaining with the base case for the 5 operation problem. The results shows that changing both the number of jobs released and also the number of operations remaining sometimes yielded better results up to a due date range of 15 days and yielded slightly worse results after that.

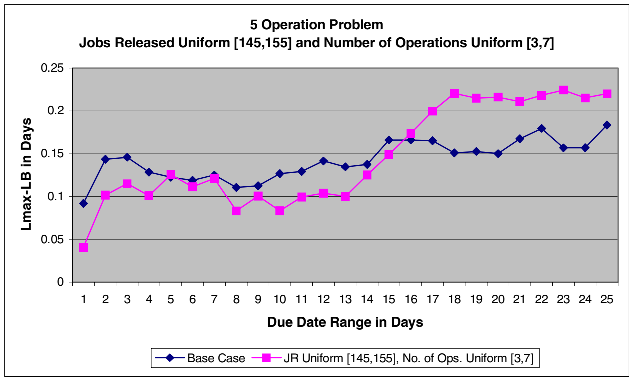

Figure 7: Varying Number of Jobs Released and the Number of Operations

5 CONCLUSIONS

The Virtual Factory has been shown to perform well in a rolling horizon setting under a variety of different conditions. The slight increase in the difference between L_{max} and LB as the total horizon length is increased could be the result of a deterioration in the quality of the scheduling solutions or the lower bound. In any case, the differences are quite small with respect to the total horizon length and the number of jobs completed.

Future experimentation will concentrate on using the rolling horizon methodology to further evaluate the multi-factory scenarios described in Thoney et al. (2002). Evaluating these scenarios in a rolling horizon setting is especially important to eliminate the many transient effects found in initial experimentation. In addition, studying alternative routing and when to release jobs into the shop in a rolling horizon setting will be carried out using the knowledge gained for transient settings in Weintraub et al. (1999) and Zozom et al. (2002), respectively.

REFERENCES

Baker, K.R. and Z. Su. 1974. Sequencing with due-dates and early start times to minimize maximum tardiness. *Naval Research Logistics Quarterly* 21 (1): 171-176.

Demirkol, E., S. Mehta, and R. Uzsoy. 1998. Benchmarks for shop scheduling problems. *European Journal of Operational Research* 109: 137-141.

Hodgson, T.J., D. Cormier, A.J. Weintraub, and A. Zozom. 1998. Satisfying due dates in large job shops. *Management Science* 44 (10): 1442-1446.

Hodgson, T.J., R.E. King, K.A. Thoney, N. Stanislaw, A.J. Weintraub, and A. Zozom. 2000. On satisfying due-dates in large job shops: idle time insertion. *IIE Transactions* 32: 177-180.

Lawrence, S.R. and T.E. Morton. 1986. Patriarch: Hierarchical production scheduling. *National Bureau of Standards Special Publication* 724: 87-97.

Thoney, K.A., T.J. Hodgson, R.E. King, M.R. Taner, and A.D. Wilson. 2002. Satisfying due-dates in large multi-factory supply chains. *IIE Transactions* 34: 803-811.

Vepsalainen, A.P.J. and T.E. Morton. 1988. Improving local priority rules with global lead-time estimates: A simulation study. *Journal of Manufacturing and Operations Management* 1*:* 102-118.

Weintraub, A.J., D. Cormier, T.J. Hodgson, R.E. King, J.R. Wilson and A. Zozom. 1999. Scheduling with alternatives: A link between process planning and scheduling. *IIE Transactions* 31 (11): 1093-1102.

Zozom, A., T.J. Hodgson, R.E. King, A.J. Weintraub and D. Cormier. Determining release times to guarantee due-date performance. Accepted by *IIE Transactions*.

AUTHOR BIOGRAPHIES

KRISTIN A. THONEY, an Assistant Professor in the Textile and Apparel, Technology and Management Dept. at NC State, joined the faculty in 2000. She earned her Ph.D. in Industrial Engineering and Operations Research and her MS in Operations Research also from NC State. Kristin's research interests include production scheduling, logistics, and supply chain modeling. She is a member of INFORMS and IIE. Her email and web addresses are <kristin_thoney@ncsu.edu> and <http://www.tx.ncsu.edu/faculty_center/directory/>.

JEFFREY A. JOINES is an Assistant Professor of Textile Engineering at NC State University. He received his B.S.I.E, B.S.E.E., M.S.I.E., and Ph.D. from NC State. He was the recipient of the 1997 Pritsker Doctoral Dissertation Award from IIE. He served as the 2000 Proceeding Editor for WSC and will be the program chair in 2005. His email and web addresses are <JeffJoines@ncsu.edu> and <http://www.tx.ncsu.edu/faculty_center/directory/>.

PADMANABHAN MANNINAGARAJAN is a graduate student in the Textile and Apparel, Technology and Management Dept. at NC State. He received a B.Tech in Textile Technology from P.S.G College of Technology in Coimbatore, India. His email address is <pmannin@unity.ncsu.edu>.

THOM J. HODGSON, the James T. Ryan Professor of Industrial Engineering and Director of the Integrated Manufacturing Systems Engineering Institute at NC State, joined the faulty in 1983 from 13 years on Univ. of Florida's Faculty and at Ford Motor Co. He earned a BSE in science engineering, a MBA in quantitative methods and a Ph.D. in industrial engineering all from University of Michigan. Thom's research interests include production scheduling, inventory control, logistics, real-time control of systems and applied operations research. He is a member of INFORMS and IIE. His email and web addresses are <hodgson@eos.ncsu.edu> and <http://www.ie.ncsu.edu/hodgson/ >.